Multilingual Stay-At-Home COVID-19 Screening Portal

Sahil Manoj Bhatt
International Institute of Information Technology Hyderabad
sahil.bhatt@research.iiit.ac.in

Abstract

The COVID-19 pandemic has spread so rapidly that even the slightest symptoms of cold and cough now frighten people. Visiting a hospital is not feasible because not only are most places shut down, but also it involves a high risk of getting infected by the virus unknowingly. In this research project, I propose an 'Online AI Doctor' which involves an AI, NLP and Big Data-based solution. The software portal proposed allows users to provide input symptoms, medical history and other details in a language of their choice, which, along with other factors would be used to make recommendations related to COVID-19 screening.

1. Introduction

The use of Artificial Intelligence in the field of healthcare is getting more and more common everyday. From identifying tumors to recommending a certain treatment method, AI has made inroads everywhere. Another field gaining a lot of momentum is Natural Language Processing, specially when it comes to recognising medical terminology. The main idea of the proposed AI-based portal is to try and eliminate the need for people to visit hospitals and clinics to get themselves screened for coronavirus.

The reason behind developing this is that there have been multiple instances wherein people visit testing centres out of fear that they might be carrying the virus, even when they are not carrying it or have no reason to worry. This is especially important given the wide spread of fake news which quite often leads people into believing false information.

Providing a safe mechanism to screen people without them having to go out of the comfort of their home will allow all 'seemingly genuine' cases to be filtered out and reduce the burden on existing healthcare facilities and also reduce delays in testing. It will also prove to be beneficial for people who do not have immediate access to healthcare facilities. Most importantly, such a feature should be available in multiple languages so that people can describe symptoms better.

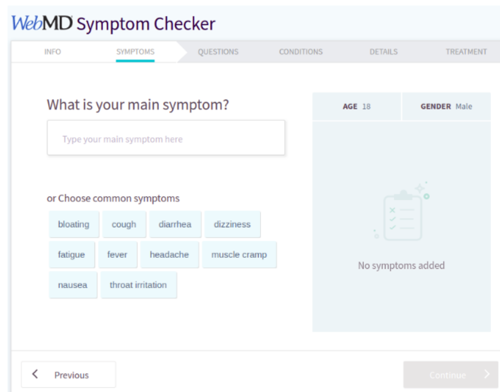

Figure 1. WebMD symptom checker interface

2. Literature Review

Several applications exist which allow people to enter their symptoms by selecting options or via a conversation with a chatbot [9]. The chatbot asks the user some relevant questions, and generates a response from the inputs given to it.

WebMD provides a Symptom Checker service that gives users options to enter details like age,sex,symptoms and medical history etc. based on which it provides a list of possible conditions that the user might have. The service is up to date and also includes Coronavirus as a possible medical condition. One can try the service here: https://symptoms.webmd.com/default.htm

Babylon Health and Mayo Clinic also have services that offers symptom checking. Babylon Health uses its chatbot in a manner similar to WebMD. It also suggests possible next steps that the user can take after seeing the results. One can try out Babylon Health's service here: https://www.babylonhealth.com/ask-babylon-chat

Neural Machine Translation (NMT) is an end-to-end learning approach for automated translation. A lot of companies are working on their AI and deep learning frameworks.For example, Google has published its work on GNMT [11](Google Neural Machine Translation) which addresses many of the issues faced by normal NMT sys-

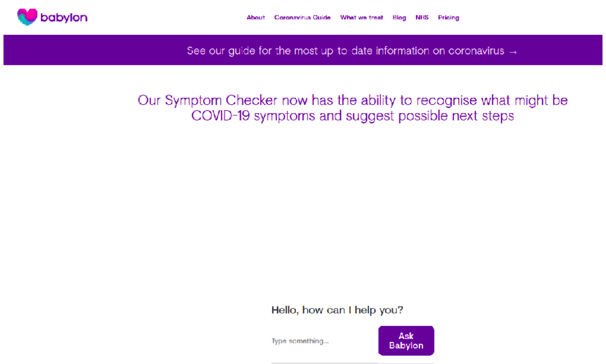

Figure 2. Babylon symptom checker interface

tems, namely computational cost and difficulty with rare words. Frameworks like TensorFlow and PyTorch also provide their own `Seq2seq`(sequence-to-sequence) modelling systems [5], which can be used to construct language translation models.

Initially, many of the AI systems used in medical diagnosis applied rule-based systems in order to solve the problem [8]. The data set used by such systems is catalogued in the form of decision tree-like structures or ”if-then-else” statements. However, using such constrained methods is not feasible as it may not be able to adapt to newer diseases and symptoms. Therefore, more recent techniques have focused on creating and using ’models’ of a disease.

3. System Architecture

The proposed portal would be designed as an application that can be used across platforms, i.e. Desktops, laptops, phones etc. The application would consist of the following key components

3.1. User Authentication

First, users would be asked to register themselves and provide their name, age, sex and location, along with a password. The login would use a secure authentication scheme, and an additional feature to include sign-up and sign-in using other platforms like Google or Facebook will also be provided. Besides this, user data would be stored in a central database. This would be used not only for keeping track of user data, but also to maintain statistics about the Screening app.

3.2. Providing Symptom Input

Users, after authentication, would be asked to choose a language of their choice (this will not include all possible languages as it is difficult to find accurate language models for all of them). The user now would be give three options:

- Choose symptoms from the options available - these options would be taken from a huge dataset of diseases and common,related symptoms
- Enter a preferably short description of symptoms faced by the user in a textbox
- Explain symptoms briefly via speech input

3.3. Translate and extract meaning from Input data

Here, we use an NMT model in order to translate our given input into English in case the original input was not in English. In case the data was provided using the set of available options, the symptoms are translated by providing it to our model (using it as a blackbox). If the data was provided using a textual description, the software will use a seq2seq model to translate the text into a sequence in English. This may then be parsed to find out all keywords that highlight symptoms using NLP-based Domain specific entity recognition techniques [6]. If the data was provided via speech, it would first be converted to text using a blackbox [7]. This text would then be used just like it would be in the previous case, wherein we directly provided textual description.

At the end of these steps, we have all available symptoms (collected and translated to English) that will be used in our next step.

3.4. Other application data and Additional Information

Besides symptoms, the user would also be asked to enter his/her medical history, either in the form of text or in the form of Electronic health records [10], if available. The appropriate filters needed to label the disease can again be parsed from the entered data.

3.5. Disease Labelling

Now, we have a list of symptoms as well as other labels [3] (like age, sex,medical history details,smoker/non-smoker, asthmatic or not etc.) needed to make an attempt to screen the user for COVID-19. These labels are given as input to our neural network [4] (blackbox). The neural network outputs data in the form of a disease and the score assigned to it, i.e. based on the existing data set and previously trained data, it will identify a list of possible diseases based on the input and also output a score, indicating how likely is it that the particular input matches the description of a given disease in the list.

The procedure does not stop here. The output list may still be very large if there is no good level of matching found with the input data. In such a case, the system asks questions further based on symptoms of the diseases in the list that were not entered to the user, through a list of options just like in the first Symptom-Input step. Based on the user response, the neural network assigns new scores and further filters out or shortlists a smaller list of possible diseases or conditions that the user maybe suffering from.

3.6. Integration with Contact Tracing applications and Location usage

During registration, an optional field will be provided where the user maybe asked if he/she allows the application to use data collected from Contact Tracing applications [2] such as Aarogya Setu (India), COVIDSafe (Australia) or TraceTogether (Singapore). The data obtained from these apps can be counted as factors in the disease labelling step and higher or lower scores maybe assigned to the chance of having COVID-19 based on it.

Another tool that can be potentially used as a factor in the disease labelling step is the prevalence of the virus in the user's area of residence. By enabling location based services, through various APIs available for tracking Coronavirus [1], the application will be able to find out the number of cases in the vicinity of the place where the user is staying. If there are a higher number of cases, a greater score could be assigned to the chance of having COVID-19 and vice-versa.

3.7. Disease database and notifying other apps and services

Now if the score assigned to COVID-19 is the highest, it represents a high probability of the user carrying the coronavirus. Although the app can only provide screening without any actual physical test, the software would recommend the user to get himself/herself tested. If found positive after proper physical screening, the software will add the user's initial input symptoms (especially new ones) to its vast dataset of diseases and related symptoms. Moreover, the symptoms input by the user in a given language would also be stored in the database in order to construct better models in the future. Not only this, if the user allowed the app to be integrated with a Contact tracing app like Aarogya Setu, then the application will also notify the Contact tracing app that the user has tested positive for the virus, and also pass on this data to the location-wise number of cases.

3.8. Action Plan

Doctors have a huge responsibility besides simply identifying diseases in patients. A major part of their responsibility also lies in finding how can the patient be treated, the preventive measures he/she can take to stop the condition from worsening etc. This Online AI doctor application is just a screening application which carries out evaluations based on the input it gets from the patients, but doesn't carry out any physical screening. So, a direct, formal diagnosis and drug-prescription cannot be given. What can be done, however, is providing a list of precautionary measures for the user to take for each of the possible medical conditions in the output list,along with a set of symptoms that are commonly observed in each case, so that if new symptoms are seen after a period of a few days, the patient can be informed accordingly. If the score assigned to a disease is really high (i.e. the probability of the user having the disease is very high), then the application will use the location data again to find out all nearby hospitals, clinics and testing centers, along with their contact numbers and display that to the user so that he/she can take the final decision whether to get a physical screening test done or not.

3.9. Understanding Use Cases

1. Authentication phase
    - Sign-Up
    - Login
    - Verification
    - Forgot password
    - Edit details
    - Store user details in central database
2. Input phase
    - Choose preferred language
    - Choose from available symptoms
    - Enter symptom description
    - Provide symptom description via Speech
    - Provide medical history and other details
3. Translation and Extraction phase
    - Translate Input Data from preferred language to English
    - Extract keywords from input data
4. Integration and Labelling phase
    - Get data from Contact Tracing app
    - Get location-based data about virus cases
    - Get location-based data about nearby hospitals and clinics
    - Feed inputs to the NN (neural network)
    - Output list of possible medical conditions to user along with their scores
5. Database addition
    - Enter new symptoms into database if user tests positive
    - Notify integrated applications, if any, about the user testing positive

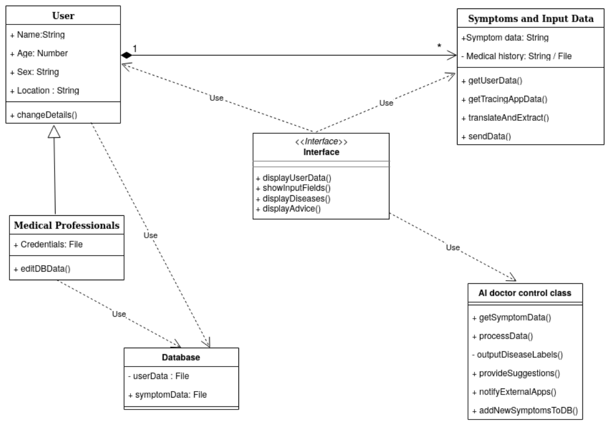

Figure 3. UML Class Diagram

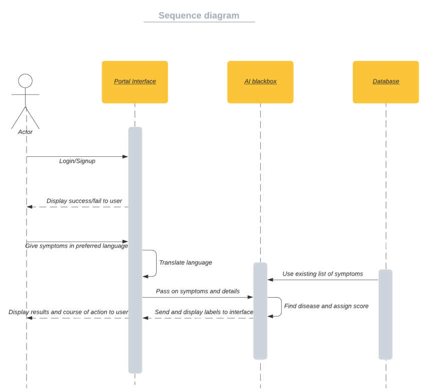

Figure 4. UML Sequence Diagram

6. Giving advice to the user
   - Displaying precautionary measures to be taken for each of the possible medical conditions from the database
   - Displaying nearby hospitals and clinics to the user

3.10. Diagrams to understand the model

The UML diagrams given on this page will help in understanding the architecture of the Screening Platform. They can also be viewed here: https://drive.google.com/drive/folders/19roIwufoL-N7O1vIWkJmksg9d2t5qXRQ?usp=sharing

4. Conclusion and Future Work

If the users of the app who were recommended to get themselves properly screened are actually tested positive, then this data could be used to understand and improve the

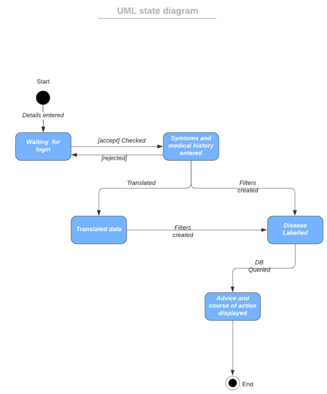

Figure 5. UML State Diagram

accuracy of the Online doctor screening portal. The threshold for recommending a user to actually get himself/herself screened can also be adjusted accordingly using patterns observed in the data collected.

This Online AI Doctor will prove to be useful nearly everywhere around the world due to flexibility in terms of language. It will also try and connect with major hospitals, clinics and medical bodies to further understand the COVID-19 situation and update datasets based on changing requirements.

References

[1] Reference: https://rapidapi.com/collection/coronavirus-covid-19. 3

[2] Anonymous. Show evidence that apps for covid-19 contact-tracing are secure and effective. *Nature.com*, 2020. 3

[3] M. BenBassat. Disease labeling via machine learning is not quite the same as medical diagnosis, 2019. Reference link:https://arxiv.org/pdf/1909.03470.pdf. 2

[4] Y. L. D.R Baughman. Classification: Fault diagnosis and feature categorization. *ScienceDirect*, 1995. 2

[5] G. Koehler. Machine translation using sequence-to-sequence learning. *Nextjournal*, 2019. 2

[6] E. Ma. How does named entity recognition help on information extraction in nlp? *Medium*, 2018. 2

[7] A. Pai. Learn how to build your own speech-to-text model. *AnalyticsVidhya*, 2019. 2

[8] W. B. S. Peter Szolovits, Ramesh S Patil. Artificial intelligence in medical diagnosis, 1988. Reference link:http://groups.csail.mit.edu/medg/people/psz/ftp/SchwartzAnnals.html. 2

[9] Z. B. Rida Sara Khan, Asad Ali Zardar. Artificial intelligence based smart doctor using decision tree algorithm, 2017. Reference link: https://arxiv.org/ftp/arxiv/papers/1808/1808.01884.pdf. 1

[10] P. E. B. S. B. Theresa A Koleck, Caitlin Dreisbach. Natural language processing of symptoms documented in free-text narratives of electronic health records: a systematic review, 2019. Reference link: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6657282/. 2

[11] Z. C. Q. V. L. M. N. W. M. M. K. Y. C. Q. G. K. M. J. K. A. S. M. J. X. L. K. S. G. Y. K. T. K. H. K. K. S. G. K. N. P. W. W. C. Y. J. S. J. R. A. R. O. V. G. C. M. H. J. D. Yonghui Wu, Mike Schuster. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. Reference link:https://arxiv.org/abs/1609.08144. 1